**友盟推送必要的配置**

**一、Xcode配置**

**1.添加依赖库**

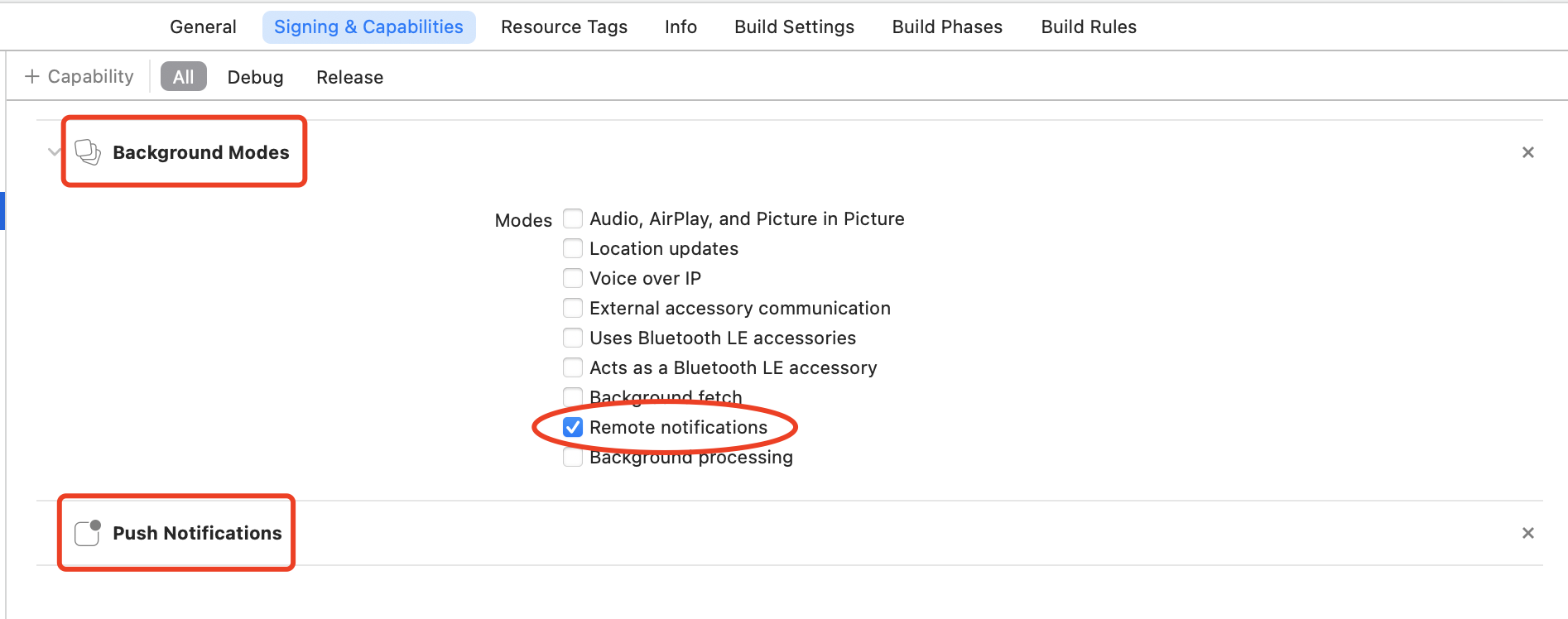
在项目设置target-> 选项卡General->Linked Frameworks and Libraries，添加以下依赖库：

|  |
| --- |
| SQL  libsqlite3.tbd  SystemConfiguration.framework  CoreTelephony.framework  libz.tbd |

**2.-ObjC设置**

Targets->BuildSettings 中 ， Other Linker Flags增加-ObjC，⼀定要添加此项，注意是⼤写C，不是⼩写c。

**3.打开推送、后台推送功能**



**4. UnityAppController.mm 文件中的代码添加**

**4.1**

didRegisterForRemoteNotificationsWithDeviceToken方法中

|  |
| --- |
| Objective-C  - (**void**)application:(UIApplication \*)application didRegisterForRemoteNotificationsWithDeviceToken:(NSData \*)deviceToken {    } |

添加以下代码：

|  |
| --- |
| Objective-C  [XHPushManager handleDeviceToken:deviceToken]; |

建议保留该方法里原来的代码，和新添加的这一行代码共存，最后看起来长这样：



该方法最终的源码如下：

|  |
| --- |
| Objective-C  - (**void**)application:(UIApplication\*)application didRegisterForRemoteNotificationsWithDeviceToken:(NSData\*)deviceToken  {  AppController\_SendNotificationWithArg(kUnityDidRegisterForRemoteNotificationsWithDeviceToken, deviceToken);  UnitySendDeviceToken(deviceToken);  [XHPushManagerhandleDeviceToken:deviceToken];  } |

要导入相关类的头文件：

|  |
| --- |
| Objective-C  #import "XHPushManager.h" |

**4.2初始化推送SDK**

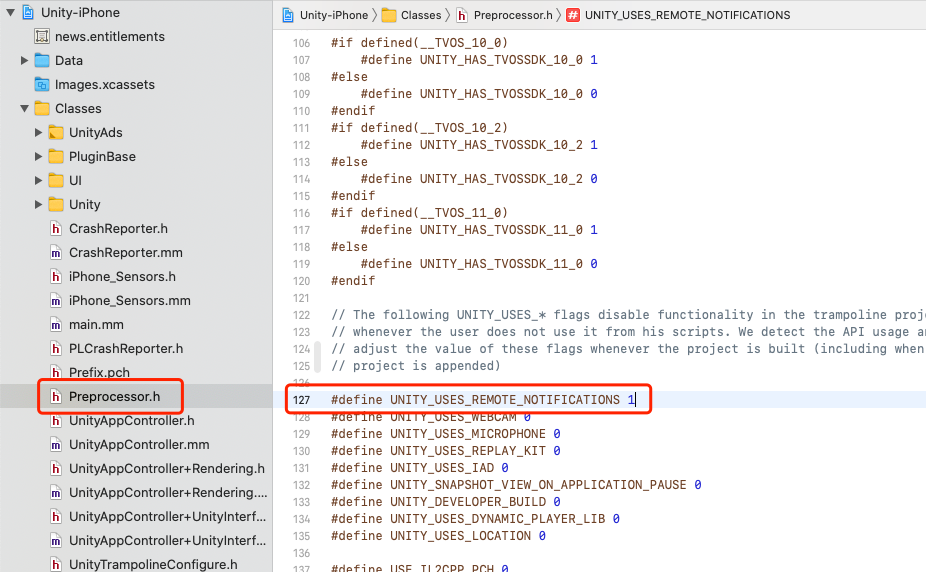
在application didFinishLaunchingWithOptions方法中，添加代码：

|  |
| --- |
| Objective-C  [[XHPushManager sharedInstance] initPush]; |

最后效果如下：

|  |
| --- |
| Objective-C  - (**BOOL**)application:(UIApplication \*)application didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions {  ...  **[[XHPushManager sharedInstance] initPush];**  ...  **return** **YES**;  } |

**5.将Preprocessor.h文件中的UNITY\_USES\_REMOTE\_NOTIFICATIONS值由0改为1，参考下图：**



这个很重要，不修改的话，接收deviceToken的系统方法将不执行，没有deviceToken服务器就不知道推给谁了。